LIVE

CREATE CHATBOT USING RASA

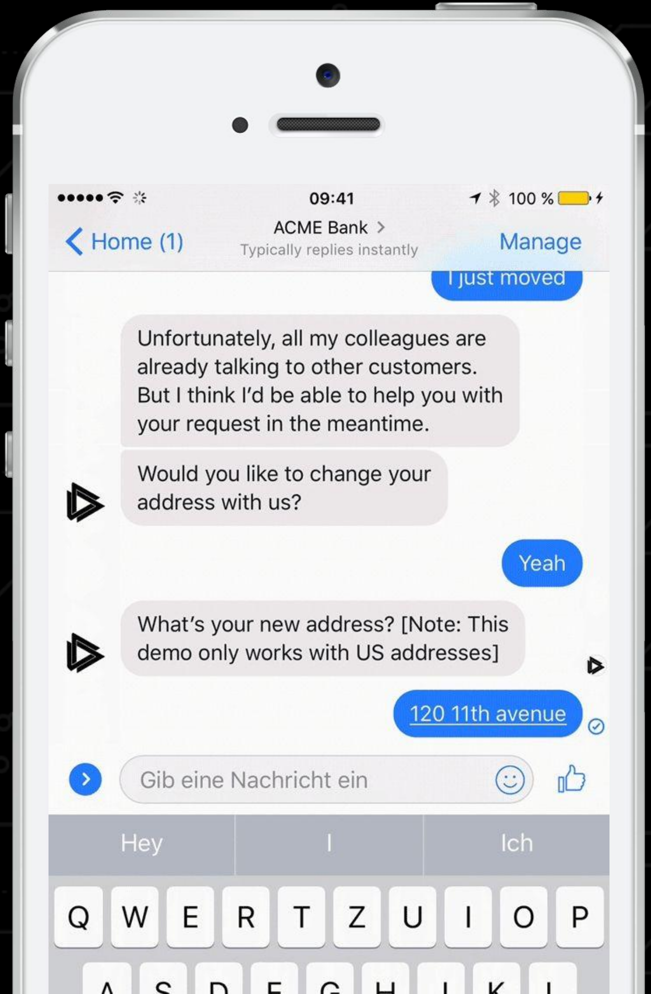
Eshaan Chauhan
Chatbot Developer

RASA Chatbot

Conversational AI with Rasa NLU & Rasa Core

Eshaan Chauhan

Conversational AI will <u>dramatically</u> change how your users interact with you.



I just moved

Unfortunately, all my colleagues are already talking to other customers. But I think I'd be able to help you with your request in the meantime.

Would you like to change your address with us?

Yeah

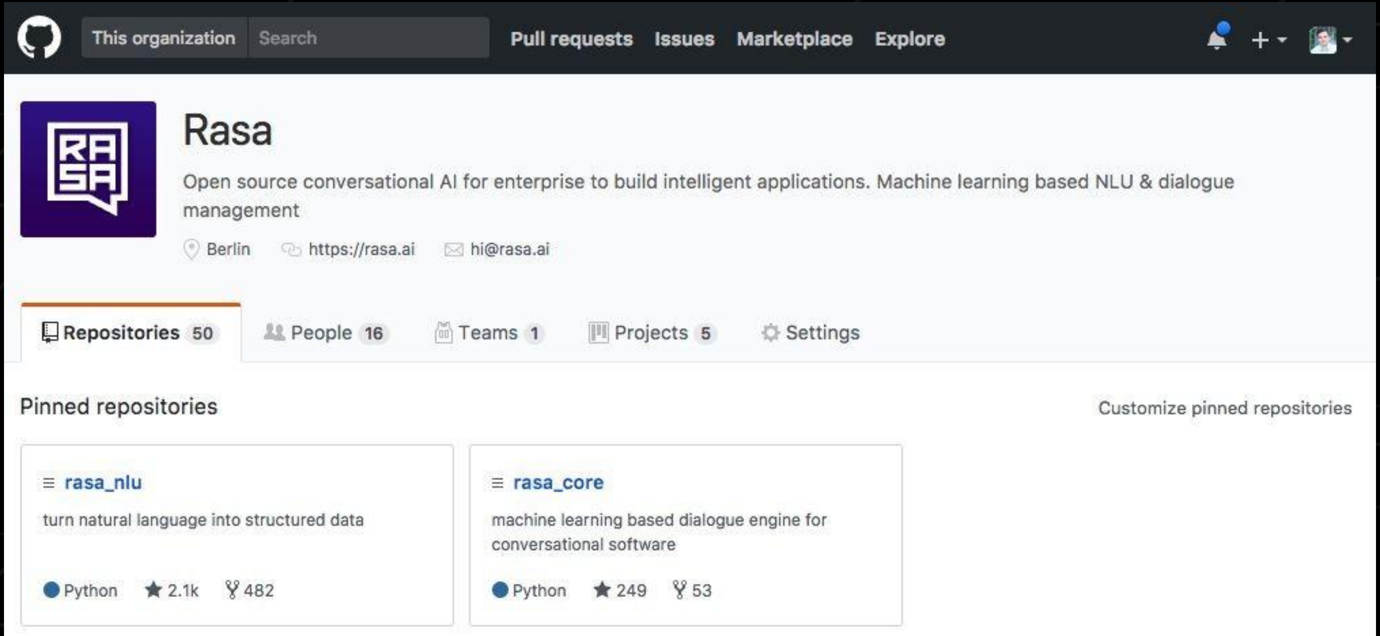
What's your new address? [Note: This demo only works with US addresses]

120 11th avenue

xamp

E

An open source, highly scalable ML framework to build conversational software



Rasa

Open source conversational AI for enterprise to build intelligent applications. Machine learning based NLU & dialogue management

Berlin https://rasa.ai hi@rasa.ai

Pinned repositories

rasa_nlu

turn natural language into structured data

Python ★ 2.1k 482

rasa_core

machine learning based dialogue engine for conversational software

Python ★ 249 53

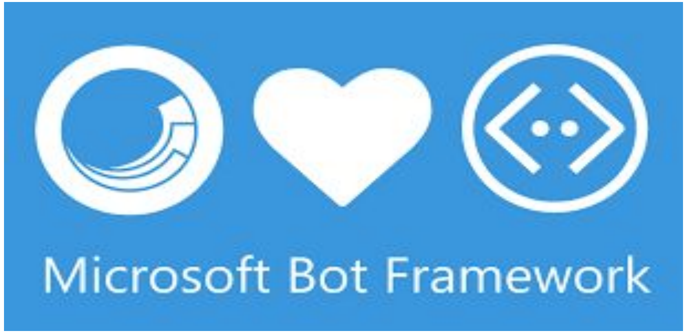
Introduction

If you want to build a conversational ai we need a bot that should be created so that a person can interact with it. Every time that a person interacts with the computer a chatbot as an assistant can guild a person on that particular website and so that is why chatbot is important.

A Conversation chatbot understands the context of the conversation and can handle any user goal gracefully and help accomplish it as best as possible. This doesn’t always mean that the bot will be able to answer all questions but it can handle the conversation well.

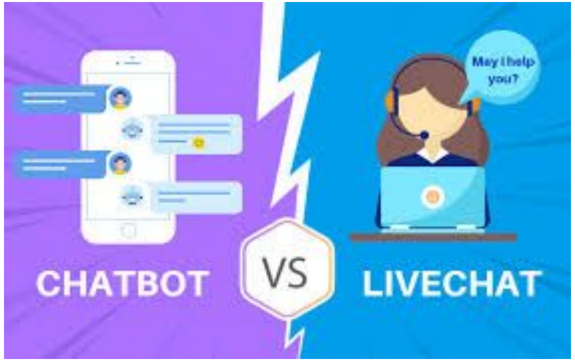
Making a chatbot or virtual assistance is used to transform the user experience. Nowadays, chatbots are gaining attraction, big or small entities such as IBM, Google, Facebook? are working on it and building their in-house products.

Different technologies which is used to make chatbots.



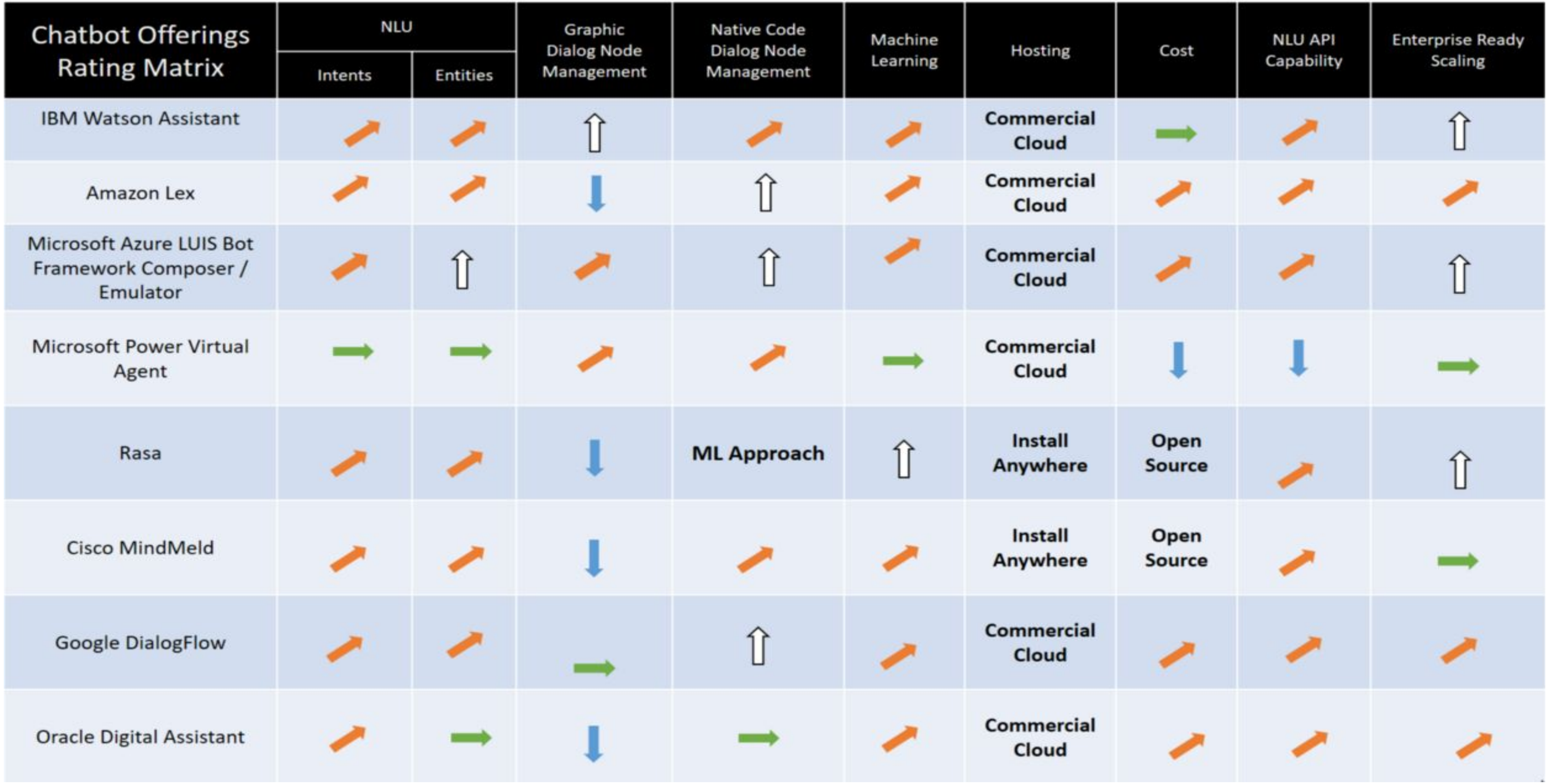
Choosing a Chatbot framework

There are several benchmarks and features to consider when choosing a chatbot framework. As your bot's user base grows, you might consider switching to another framework or using multiple frameworks for different iterations of the same bot. Using your chosen framework's platform integration tools, you might also deploy your bot across more than one platform. Although it's great to try out different bot-building tools and to launch your bot in multiple places, there's one thing that's easy to overlook—consistency.



Comparing different Chatbot Software Together

⇧ Exceptional ↗ Good → Average ↓ Poor



Chatbot Offerings Rating Matrix	NLU		Graphic Dialog Node Management	Native Code Dialog Node Management	Machine Learning	Hosting	Cost	NLU API Capability	Enterprise Ready Scaling
	Intents	Entities							
IBM Watson Assistant	↗	↗	⇧	↗	↗	Commercial Cloud	→	↗	⇧
Amazon Lex	↗	↗	↓	⇧	↗	Commercial Cloud	↗	↗	↗
Microsoft Azure LUIS Bot Framework Composer / Emulator	↗	⇧	↗	⇧	↗	Commercial Cloud	↗	↗	⇧
Microsoft Power Virtual Agent	→	→	↗	↗	→	Commercial Cloud	↓	↓	→
Rasa	↗	↗	↓	ML Approach	⇧	Install Anywhere	Open Source	↗	⇧
Cisco MindMeld	↗	↗	↓	↗	↗	Install Anywhere	Open Source	↗	→
Google DialogFlow	↗	↗	→	⇧	↗	Commercial Cloud	↗	↗	↗
Oracle Digital Assistant	↗	→	↓	→	↗	Commercial Cloud	↗	↗	↗

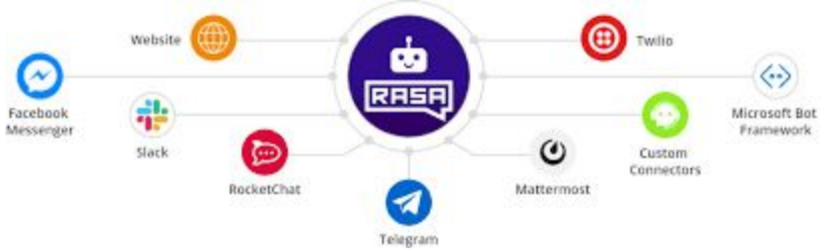
NLU,NLU and NLG

I. NLP, or **Natural Language Processing** is a blanket term used to describe a machine's ability to ingest what is said to it, break it down, comprehend its meaning, determine appropriate action, and respond back in a language the user will understand.

II. NLU, or **Natural Language Understanding** is a subset of NLP that deals with the much narrower, but equally important facet of how to best handle unstructured inputs and convert them into a structured form that a machine can understand and act upon. While humans are able to effortlessly handle mispronunciations, swapped words, contractions, colloquialisms, and other quirks, machines are less adept at handling unpredictable inputs.

III. NLG, or **Natural Language Generation**, simply put, is what happens when computers write language. NLG processes turn structured data into text.

RASA Framework



Rasa is an open-source framework that has 2 major components: Rasa NLU and Rasa Core.
Rasa Stack is actually leading in the open-source machine learning toolkits that help developers create better AI chatbots with minimal training data. Why rasa is the best framework to be used to make a chatbot because -

Pros:

1. It can be deployed on your own server by keeping all the components in-house.
2. Highly customizable, so it allows developers to create a chatbot with desired features.
3. It allows multiple environments for development, staging and production.
4. Rasa offers analytics for various data that allows you to understand customers better.
5. This AI chatbot framework works on the basis of interactive learning,
6. hence it keeps learning on its own as it interacts with people.

Cons:
Rasa isn’t exactly suitable for beginners so new developers could be a little intimidated by this AI chatbot framework.

Integrate with:

Facebook Messenger
Rocket.Chat
Slack
Telegram
Twilio

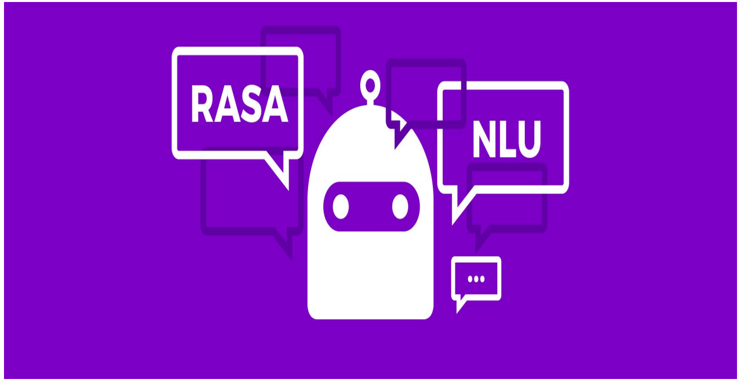
What is Rasa?

Rasa is a machine learning conversational ai which is used to build a chatbot. It is one of the best open-source machine learning toolkits which is used to developer complex chatbot with minimal training data.

It is based on two frameworks-

1.Rasa nlu– a library for natural language understanding (NLU) which does the classification of intent and extracts the entity from the user input.

2.Rasa Core– A chatbot framework that predicts the next best response or action based on conversational history.



I

Why Rasa?

Runs Locally

- No Network Overhead
- Control QoS
- Deploy anywhere

Own Your Data

- Don’t hand data over to big tech co’s
- Avoid vendor lock-in

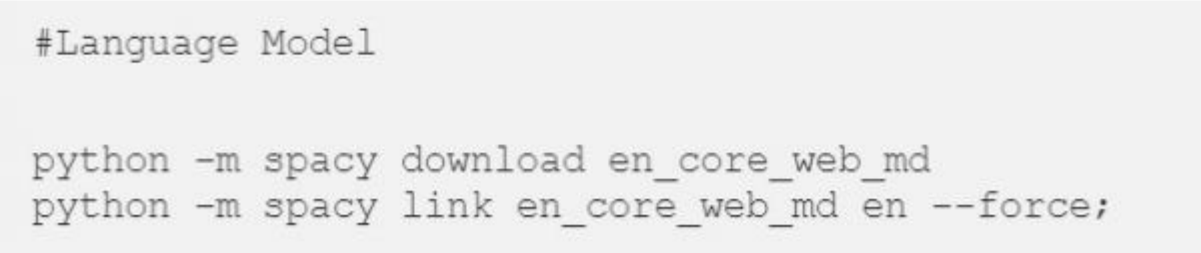
Hackable

- Tune models for your use case

Setup

1.Rasa Install (pip install rasa)

2.You also need to install a spaCy English language model:



```
#Language Model


python -m spacy download en_core_web_md
python -m spacy link en_core_web_md en --force;
```

3.Create a empty folder where you will put all your custom rasa files

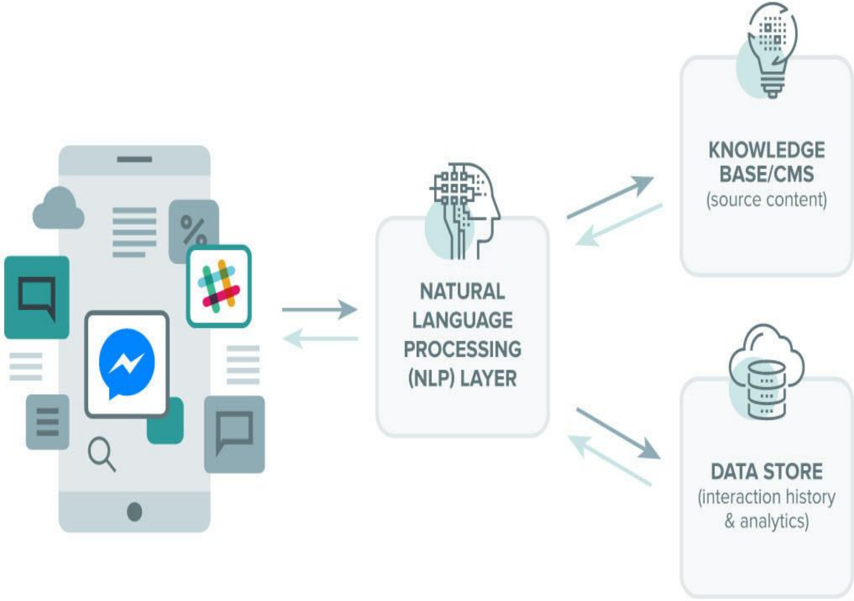
4.In the command line or terminal write rasa init where it will create the necessary files for the chatbot.

5.Then we train the model using rasa train

6. Then we run the model using rasa shell

How Do Chatbots Work?

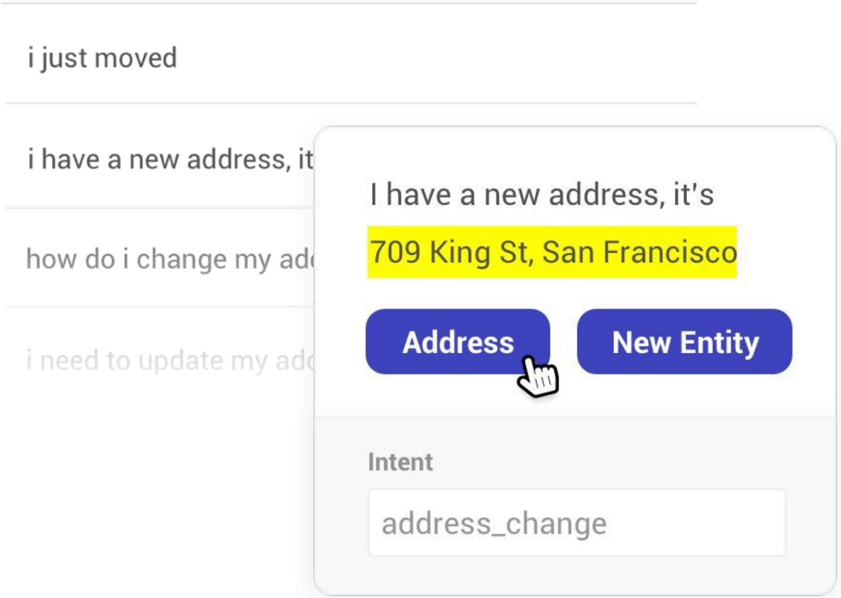
Chatbots process the text presented to them by the user (a process known as “parsing”), before responding according to a complex series of algorithms that interprets and identifies what the user said, infers what they mean and/or want, and determine a series of appropriate responses based on this information.


NATURAL LANGUAGE PROCESSING (NLP) LAYER
KNOWLEDGE BASE/CMS
(source content)
DATA STORE
(interaction history & analytics)

Rasa NLU: Natural Language Understanding

Goal: create structured data

I have a new address, it’s 709 King St, San Francisco



i just moved

i have a new address, it

how do i change my ad

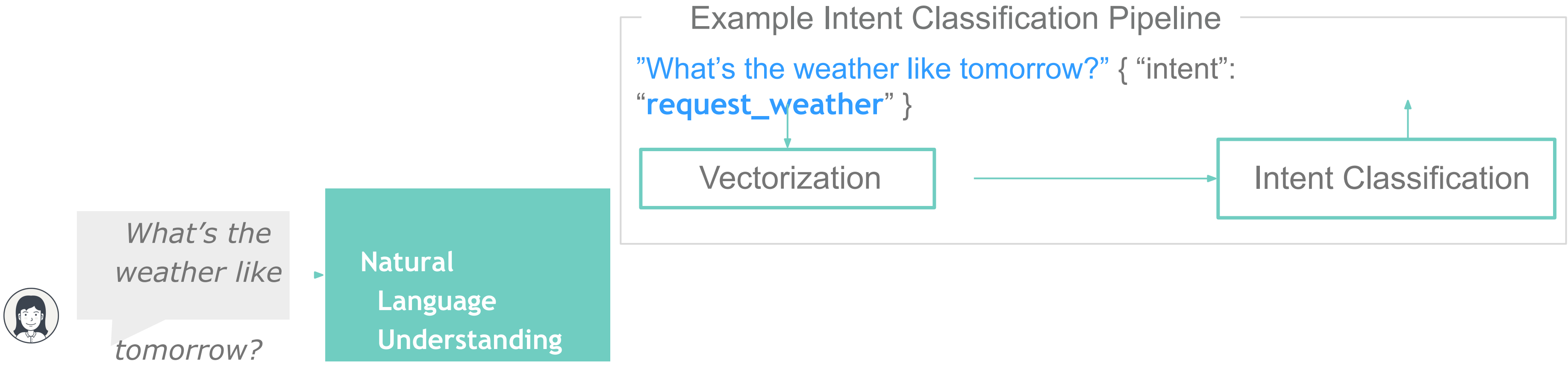
i need to update my ad

I have a new address, it's 709 King St, San Francisco

Intent

address_change

Rasa NLU: Natural Language Understanding



What's the weather like tomorrow?

Natural Language Understanding

Example Intent Classification Pipeline

"What's the weather like tomorrow?" { "intent": "**request_weather**" }

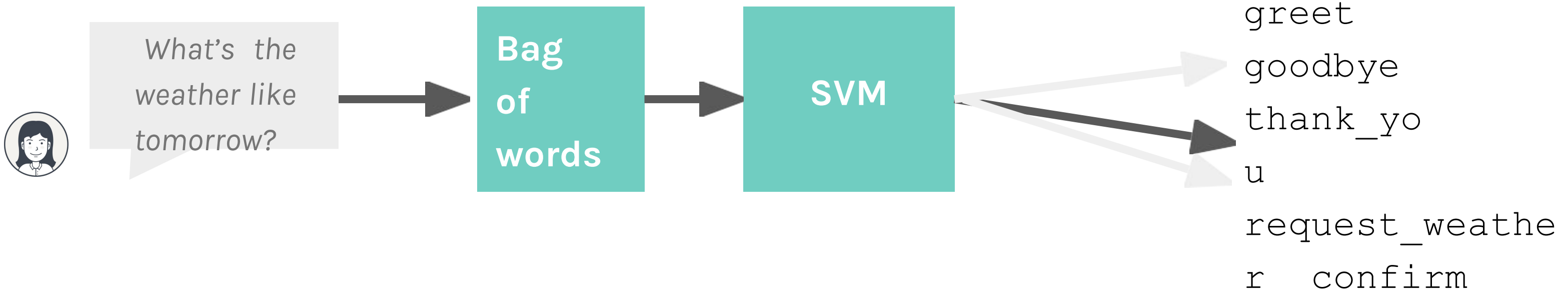
Vectorization

Intent Classification

Rasa NLU: Natural Language Understanding

Bags are your friend

$$\{v_1, \ldots, v_s\} \rightarrow \frac{1}{s}\sum_i v_i$$



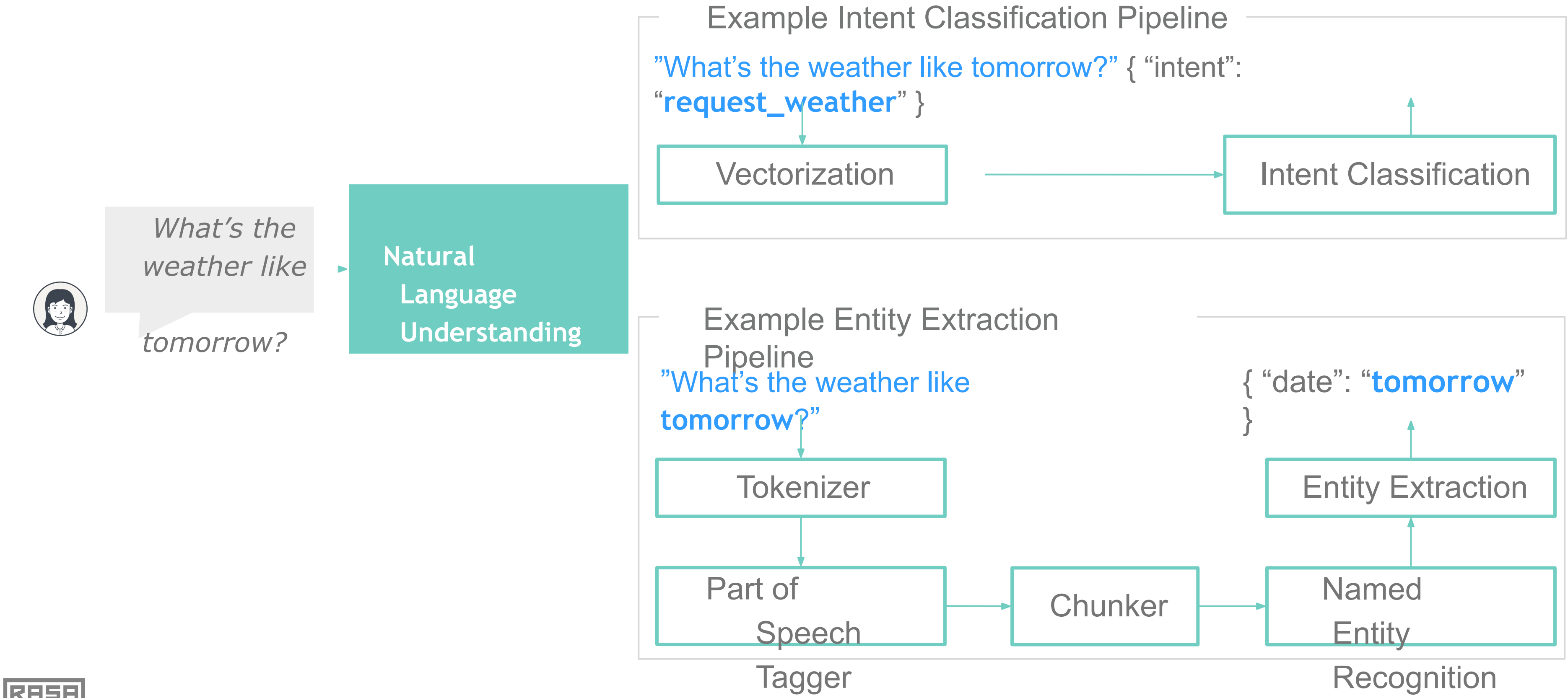
What's the weather like tomorrow?

Bag of words

SVM

greet
goodbye
thank_yo
u
request_weathe
r confirm

Rasa NLU: Natural Language Understanding



What's the weather like tomorrow?

Natural Language Understanding

Example Intent Classification Pipeline

"What's the weather like tomorrow?" { "intent": "**request_weather**" }

Vectorization

Intent Classification

Example Entity Extraction Pipeline

"What's the weather like **tomorrow**?"

{ "date": "**tomorrow**" }

Tokenizer

Part of Speech Tagger

Chunker

Named Entity Recognition

Entity Extraction

What are different parts in Rasa

What is Intent?

It is the verb in your dialog which you write as your user input.

“I want to order a book” or “Placing order for a given book”. Here, the intent would be “ordering”.

Example-Good morning .Hi, Hello, hey there

Intent : greet

What is Entity?

It is the nouns in the dialog as it is an useful information from the user input that can be extracted.

Example: Reserve a table at a hotel tomorrow night

Intent : book a table

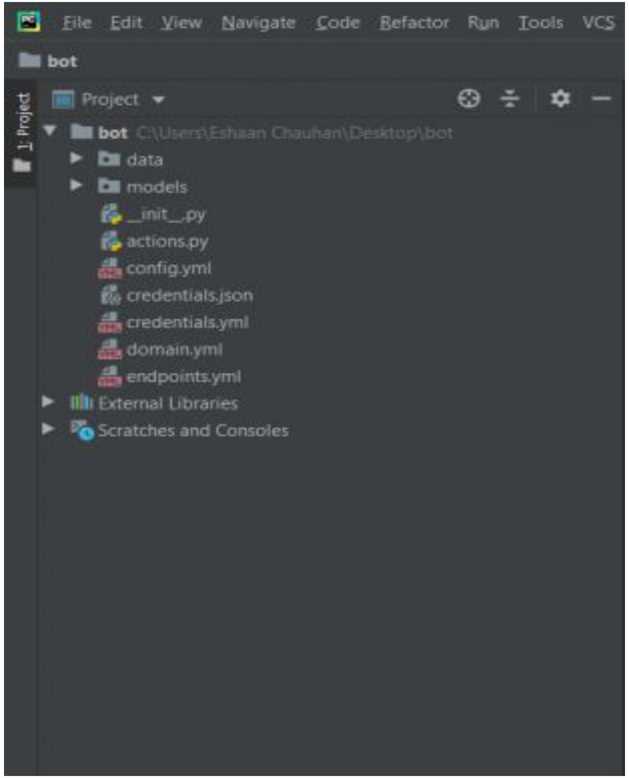
Entity: Place: hotel

time: tomorrow night

What are Stories?

These are sample interaction between the user and bot, defined in terms of intent captured and actions performed. So developer can mention what to do if you get a use input of some intent with/without some entities.

Different Files in RASA



1.You need to know how to make a chatbot using rasa.You will get the following files while you make the chatbot using rasa init command.

(first you need to install rasa before creating the chatbot)

Part of rasa folder

Intent and entity ——————> nlu.md

Story(Dialogue Management)——->Stories.md

Actual output———————>domains.yml

Custom response———————->actions.py

Different Files in RASA

nlu.md

It is the rasa nlu where you train your bot for all the actions that you want It to perform in your chatbot.

Nlu capture the intent and the training data for the intent.

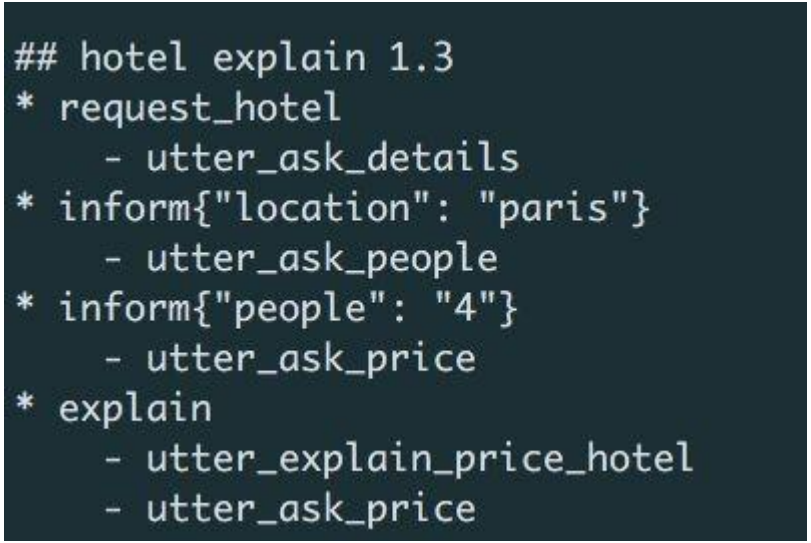
As you feed in more intent then with each data that you put you need to retrain the rasa chatbot by using the command rasa train

Intent has all the feature inside it so when you ask a question it is actually the intent which has the given question inside it.

```
## intent:greet
- good morning
- good evening
## intent:good
- i m good.
- good. thanks for asking
- Good.
- Gud
- guds
- gooood
- i am great
## intent:hello_world
- world
- programming
- first time
- search
```

STORIES.md

```
## happy path
* greet
- utter_greet
* mood_great
- utter_happy
```



```
## hotel explain 1.3
* request_hotel
    - utter_ask_details
* inform{"location": "paris"}
    - utter_ask_people
* inform{"people": "4"}
    - utter_ask_price
* explain
    - utter_explain_price_hotel
    - utter_ask_price
```

Stories.md is a file which show the interaction between the user and the bot.

Take

1.There are two sub-points of greet and mood great. You can see that the intent which was given in nlu.md is been referred here. So when you ask a question as hi it will great how are you that is because in domain the utter_great function is defined as “How are you “ so whenever you input a value which is present in intent as hi,hello,good morning etc you will get a fixed response of “Hi how are you”.

The stories help the user to create set stories which will follow the path which is been written in the given stories.md.

When you start your bot you will ask hello and with this intent it will go to the stories.md happy path section where it will greet and the required response *greet (utter_great) will give a response as “Hey! How are you doing ?”

Domain

```
intents:
- greet
- goodbye
- affirm
- deny
- mood great
- mood_unhappy
- bot_challenge
- hello_world
- more info
- question
- good
- gr
- sendemail
- ask email
- topic wise
- search_content
- introductory
-
machine learning
- deep learning
- search_topic
- hgreet
entities:
- document
- email
```

Domain.yml

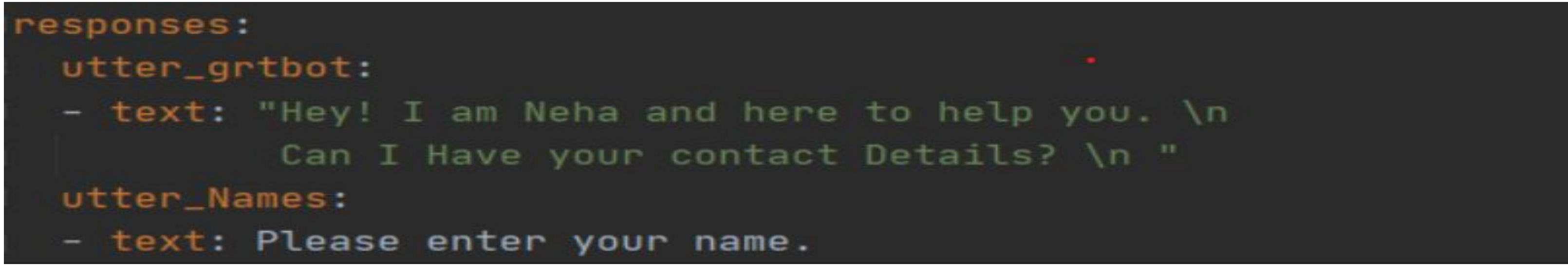
To combine nlu and stories.md we use domain.yml.

1.First all the intents which you have written in nlu.md file should be written here.

2.entities and slots will be written for the document as it is used to connect the google account and get the required files from google drive.

3.Responses -are all the intent response for a particular question which is been written in nlu.md.

All these are text and should be written in double quotes so that you get the required response in the given chatbot.

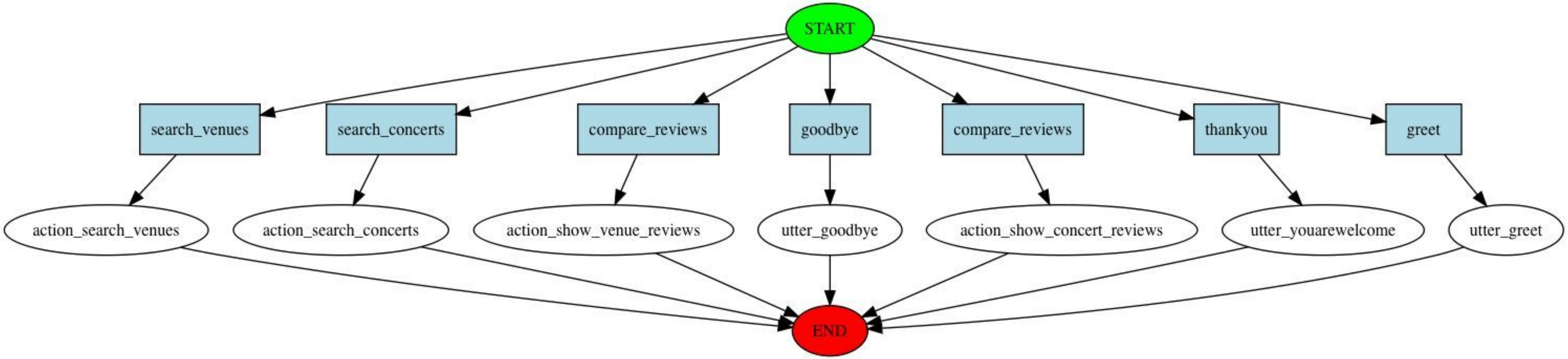


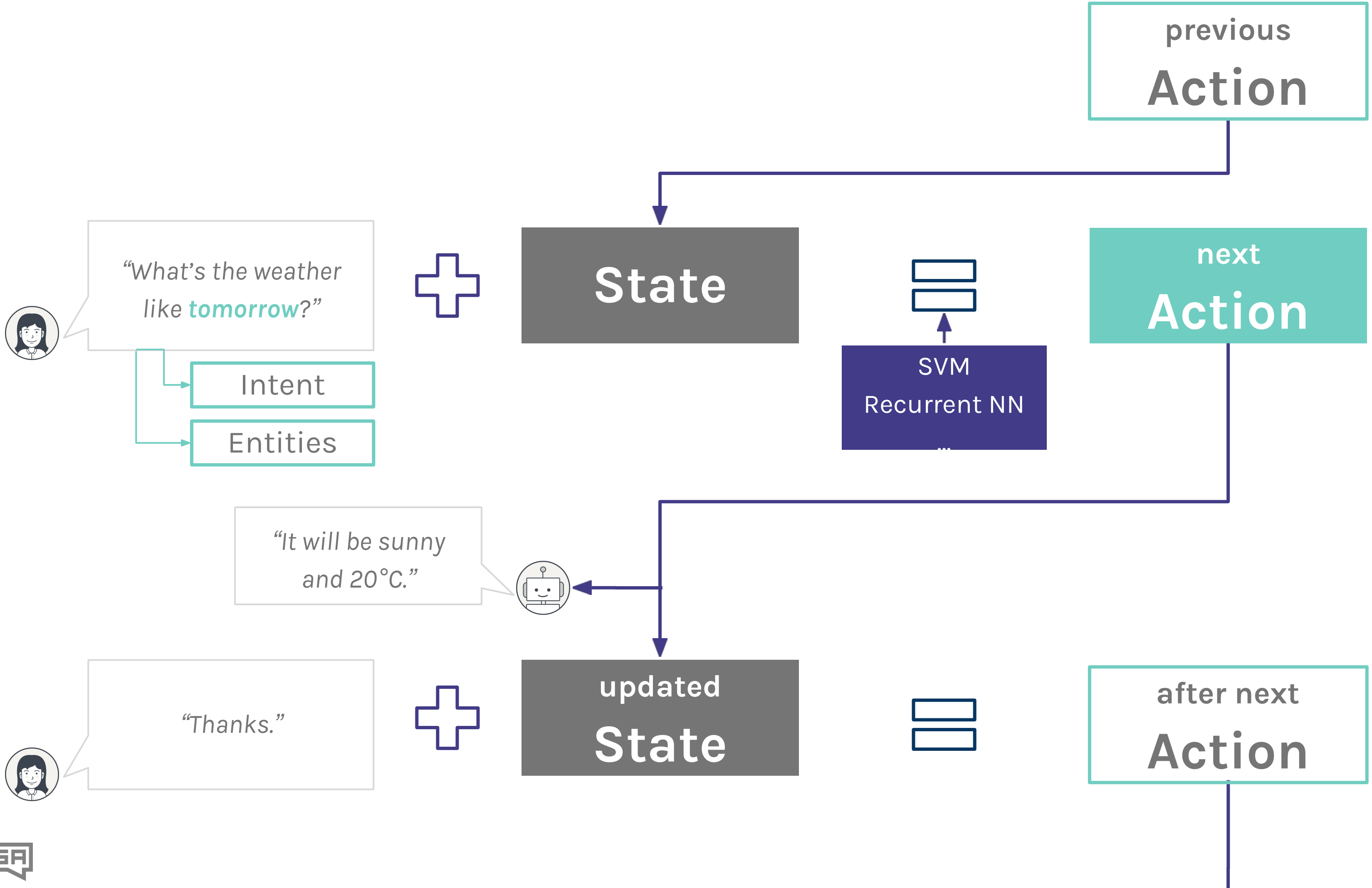
```
responses:
  utter_grtbot:
  - text: "Hey! I am Neha and here to help you. \n
          Can I Have your contact Details? \n "
  utter_Names:
  - text: Please enter your name.
```

Why Dialogue Handling with Rasa Core?

- No more state machines!
- Reinforcement Learning: too much data, reward functions...
- Need a simple solution for everyone

Why Machine Learning?





previous Action

"What's the weather like **tomorrow**?"

Intent

Entities

State

SVM

Recurrent NN

...

next Action

"It will be sunny and 20°C."

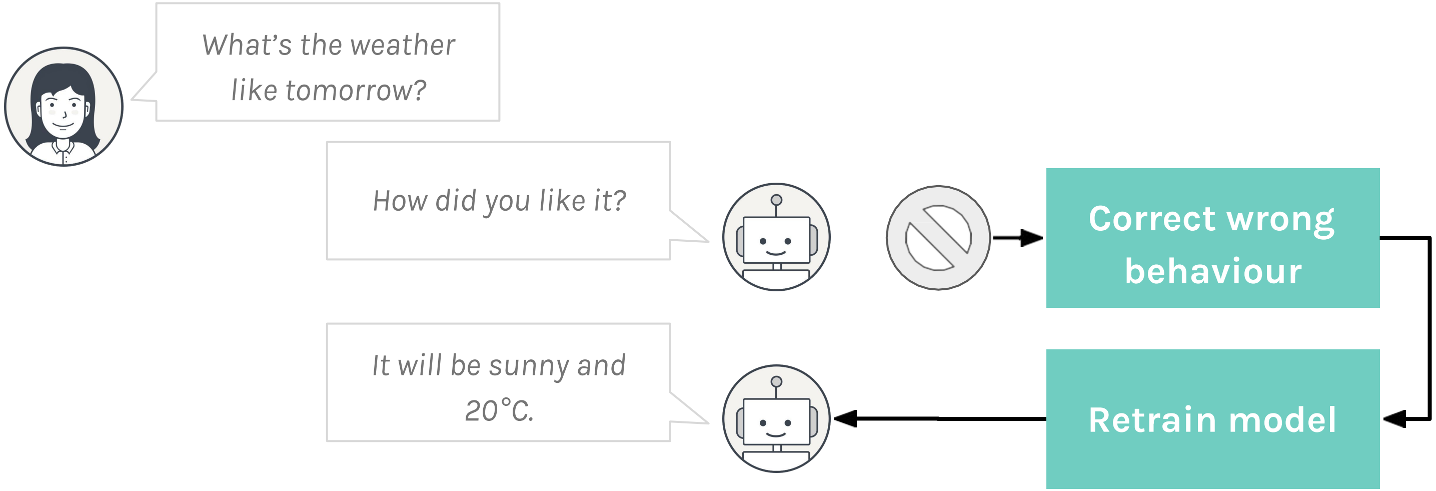
"Thanks."

updated State

after next Action

Rasa Core: Dialogue Training

Issue: How to get started? → Online Learning



What's the weather like tomorrow?

How did you like it?

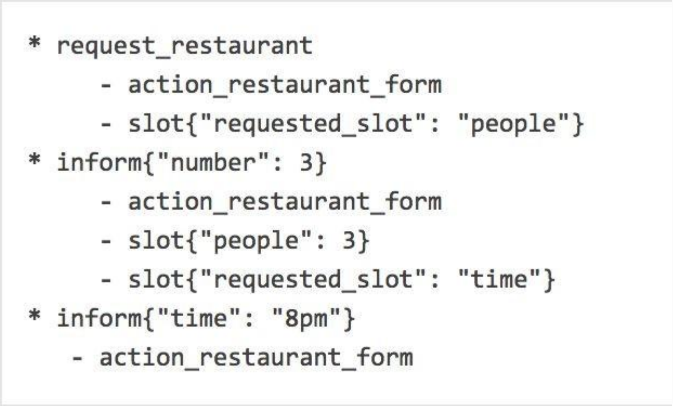
Correct wrong behaviour

Retrain model

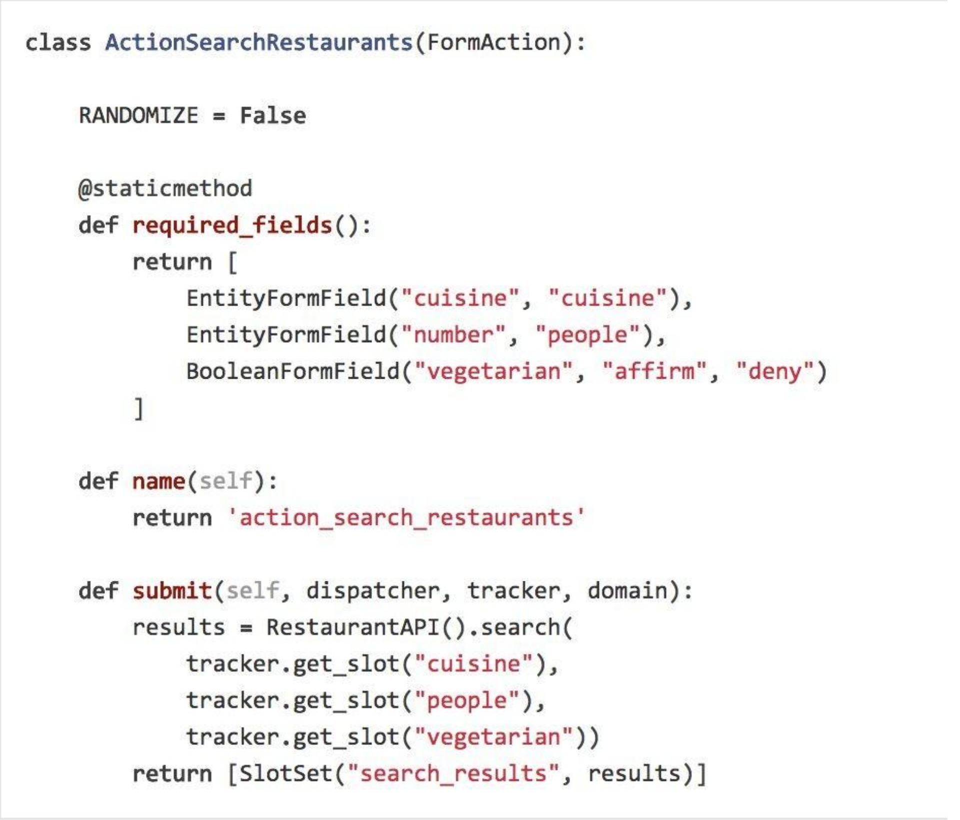
It will be sunny and 20°C.

Form Filling

- Make some form logic deterministic
- Request all fields and then call API
- Simplifies action space



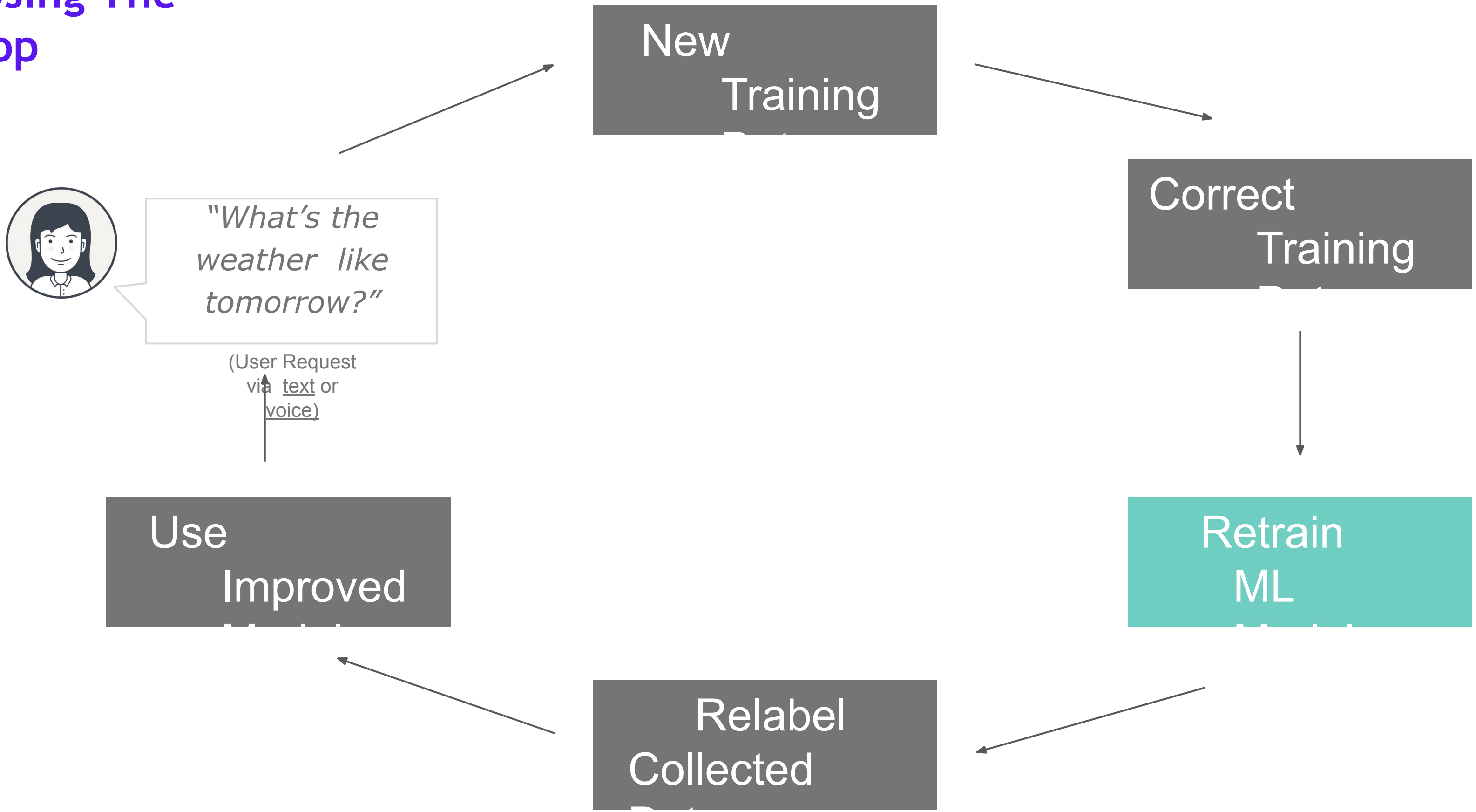
```
* request_restaurant
    - action_restaurant_form
    - slot{"requested_slot": "people"}
* inform{"number": 3}
    - action_restaurant_form
    - slot{"people": 3}
    - slot{"requested_slot": "time"}
* inform{"time": "8pm"}
  - action_restaurant_form
```



```
class ActionSearchRestaurants(FormAction):

    RANDOMIZE = False

    @staticmethod
    def required_fields():
        return [
            EntityFormField("cuisine", "cuisine"),
            EntityFormField("number", "people"),
            BooleanFormField("vegetarian", "affirm", "deny")
        ]

    def name(self):
        return 'action_search_restaurants'

    def submit(self, dispatcher, tracker, domain):
        results = RestaurantAPI().search(
            tracker.get_slot("cuisine"),
            tracker.get_slot("people"),
            tracker.get_slot("vegetarian"))
        return [SlotSet("search_results", results)]
```

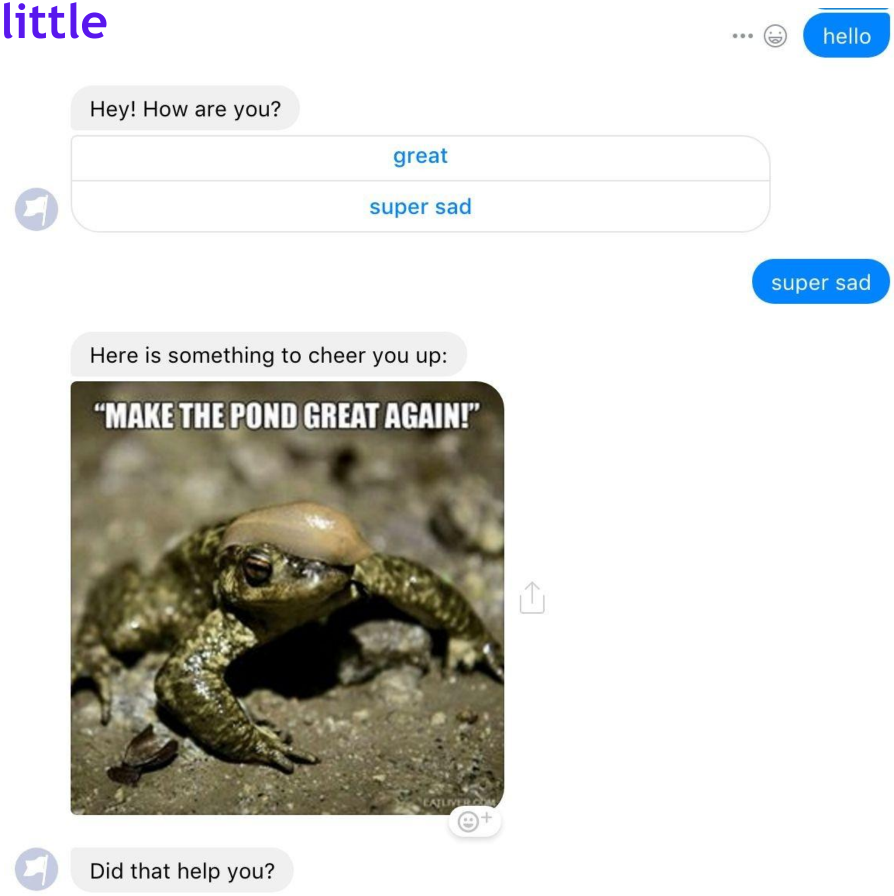
Final Thoughts

Closing The Loop


New Training
Correct Training
Retrain ML
Relabel Collected
Use Improved
“What’s the weather like tomorrow?”
(User Request via text or voice)

Under the Hood

Let’s build a fun little bot (RASA X)



hello

Hey! How are you?

great

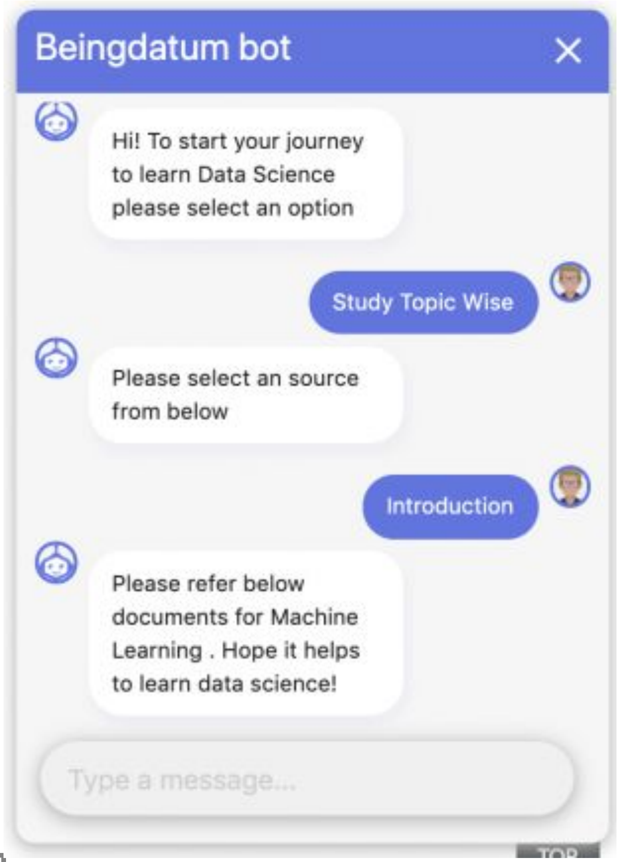
super sad

super sad

Here is something to cheer you up:

Did that help you?

Your ChatBot will Look Like



Hi! To start your journey to learn Data Science please select an option

Study Topic Wise

Please select an source from below

Introduction

Please refer below documents for Machine Learning . Hope it helps to learn data science!

Now you can start the bot by following commands-

```
rasa train
```

```
rasa run -m models –enable-api –cors “*” –ssl-certificate <> –ssl-keyfile /<> –debug
```

In the above command *–enable-api –cors “*” –ssl-certificate <> –ssl-keyfile /<> –debug* is optional which is needed if you want to deploy the bot over a https ssl website.

Thank You

Eshaan Chauhan